Input Synchronization in Low Power CMOS Arithmetic Circuit Design

Charles A. Fabian
fabian@cs.ucla.edu

Miloš D. Ercegovac
milos@cs.ucla.edu

Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90025

Abstract

Power dissipation in static CMOS circuits can be directly related to the signal transition activity of the circuit. Spurious, unwanted, transition activity can account for a large percentage of the overall transition activity. One cause of spurious activity is relative skew in the arrival time of asynchronous input signals. We measure the effects of input signal arrival skew on a typical CMOS full adder cell, on an 8-bit ripple carry, and on small partial product reduction arrays. We use three-state buffers to synchronize the inputs to these circuits and examine the trade-off.

1. Overview

Increasingly, power consumption is as an important parameter as speed and area in the design of digital integrated circuits. This trend is being driven by both the increase in portable applications as well as problems of heat dissipation associated with high integration circuits [3].

In static CMOS, signal transition activity, or *switching activity*, is directly related to the dynamic power dissipation [7], which is the most significant component of the circuit's total power dissipation. Spurious transitions, or glitching, results from imbalances in signal propagation paths and may comprise a significant percentage of the total switching activity [1]. Synchronization and self-timing techniques may be used to reduce the amount of spurious transitions [4] [5].

For a particular transition, the total energy may be computed as

$$E = P_{\text{avg}}\Delta t = \int P_{\text{inst}}\text{dt}$$

Unlike P_{avg}, E is independent of the cycle time.

In this paper, we examine the increase in switching activity that results from *input signal skew*. When analyzing the transition activity of a circuit module, it is common to assume co-incident arrival of all input signals. This assumption, which greatly simplifies the analysis or simulation, is only valid when the module's inputs are synchronized, such as through latching; however, in many cases the immediate inputs are not latched or synchronized in any manner. Without input synchronization, a degree of skew can exist between the arrival times of different input signals during the same input event. Even in cases when the inputs are latched, but the latches are physically separated from the circuit, some degree of skew can result from the distance the signals must travel. This skew can cause spurious transitions in the circuit, increasing the switching activity of the circuit. As we demonstrate, ignoring the spurious transitions that result from the relative-skew between input signals may dramatically effect the accuracy of the power consumption estimates.

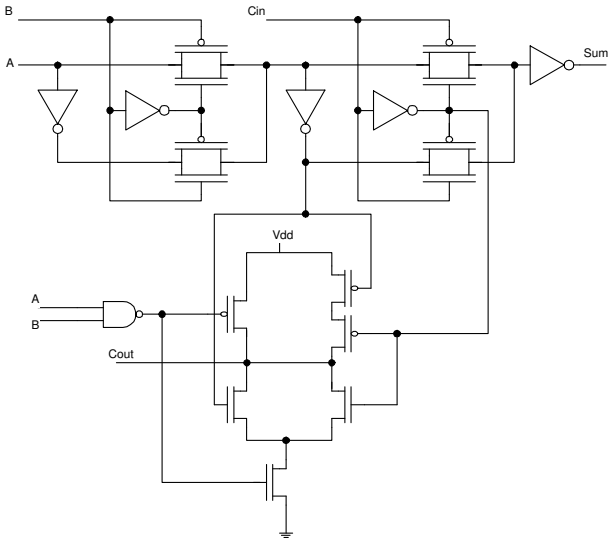

Figure 1. CMOS Full Adder

Beginning at the device level, we characterize a typical CMOS full-adder cell and analyze how its power dissipation

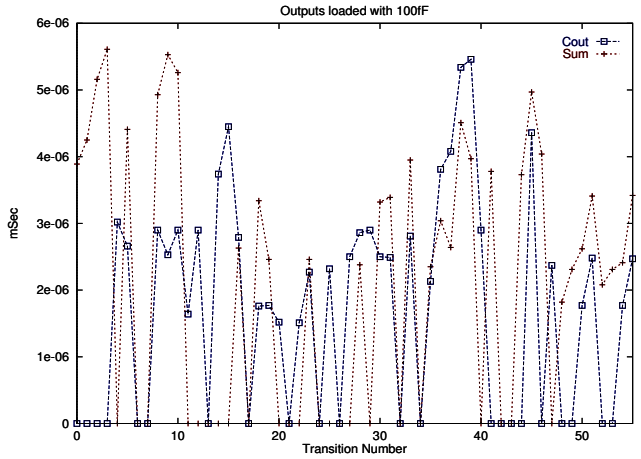

Figure 2. Full Adder: Propagation Delay by input transition.

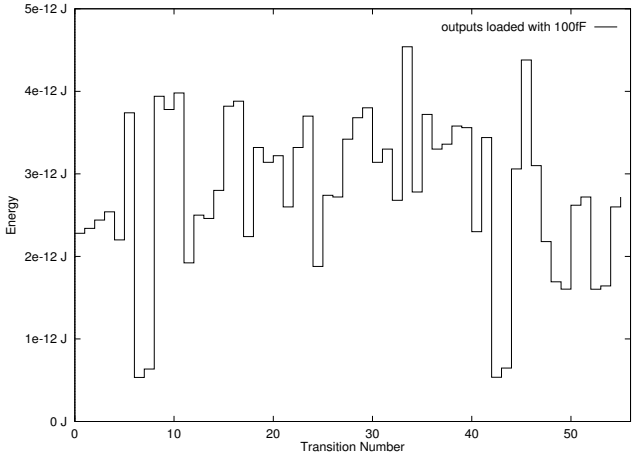

Figure 3. Full Adder: Total Energy by input transition.

is affected by the relative skew of its input signal arrival times. We then simulate a simple 8-bit ripple carry adder and a small partial product reduction array to demonstrate the effect of input signal arrival skew on its switching activity.

2. Full Adder Cell Characterization

We used HSpice [6] to characterize the power dissipation and delay profile of the CMOS full adder cell from the Berkeley Low-Power library. (Circuit diagram in figure 1.) We simulated with parameters for the MOSIS 1.2μm process. Assuming co-incident input signal arrival (0 skew), we measured the power dissipation and signal delay for each of the 56 non-trivial combinations of input signal transitions (table 1). Figure 2 plots the propagation delays for the S and C_{out} signals when driving output loads of 100fF (equivalent to 5 inverters — a full load is 6 inverters); figure 3 plots the total energy consumed by the full adder for each transition. Both the power and propagation delay for this cell varies greatly according to the particular transition. Table 2 summarizes the disparity between the best and worst case delay for the cell.

	C_{out}		S	
	best	worst	best	worst
from A,B	1.6ns	5.4ns	2.4ns	5.6ns
from C_{in}	1.7ns	2.4ns	1.8ns	3.4ns

Table 2. Full Adder: Best and Worst Case Propagation Delays.

We then varied the relative skew between the A and B input signal arrival times and held the C_{in} input constant; the resulting energy consumption for selected transitions where $C_{\text{in}} = 0$ is plotted in figure 4. (The plot for $C_{\text{in}} = 1$ is similar.) For reference, the propagation delay from the C_{in} to the C_{out} signal under the same conditions is between 1.7ns, (best-case) and 2.4ns (worst-case). These results are not surprising: the mininum energy is consumed when the A and B signals arrive coincidentally. For $|\tau| \geq$ 1FA, the energy is equivalent to that of two separate transitions, as would be expected. The increase in energy as $\tau \rightarrow \pm\infty$ can be up to a factor of 20, depending upon the particular transition.

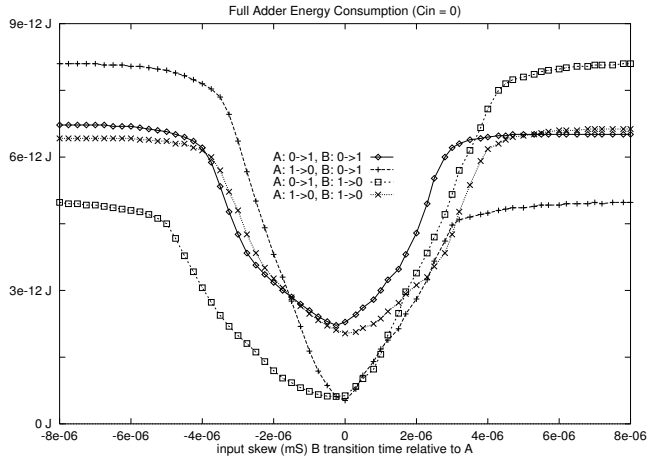

Figure 4. Energy vs. Input Skew ($C_{\text{in}} = 1$)

We performed similar simulations for the input transition combinations where the C_{in} input is also changing; the results for selected cases are plotted in figure 5. Depending upon the particular transition, the minimum power is dis-

Transition	A	B	C_{in}	Transition	A	B	C_{in}	Transition	A	B	C_{in}
0	$0 \rightarrow 1$	0	0	20	1	$0 \rightarrow 1$	$0 \rightarrow 1$	40	$0 \rightarrow 1$	$0 \rightarrow 1$	0
1	$1 \rightarrow 0$	0	0	21	$1 \rightarrow 0$	1	$0 \rightarrow 1$	41	1	$1 \rightarrow 0$	1
2	0	$0 \rightarrow 1$	0	22	$0 \rightarrow 1$	1	$0 \rightarrow 1$	42	$1 \rightarrow 0$	$0 \rightarrow 1$	1
3	0	$1 \rightarrow 0$	0	23	$1 \rightarrow 0$	$1 \rightarrow 0$	$0 \rightarrow 1$	43	$0 \rightarrow 1$	$1 \rightarrow 0$	1
4	$0 \rightarrow 1$	$0 \rightarrow 1$	0	24	$0 \rightarrow 1$	0	$1 \rightarrow 0$	44	1	$0 \rightarrow 1$	1
5	1	$1 \rightarrow 0$	0	25	$1 \rightarrow 0$	0	$1 \rightarrow 0$	45	$1 \rightarrow 0$	1	1
6	$1 \rightarrow 0$	$0 \rightarrow 1$	0	26	0	$0 \rightarrow 1$	$1 \rightarrow 0$	46	$0 \rightarrow 1$	1	1
7	$0 \rightarrow 1$	$1 \rightarrow 0$	0	27	0	$1 \rightarrow 0$	$1 \rightarrow 0$	47	$1 \rightarrow 0$	$1 \rightarrow 0$	1
8	1	$0 \rightarrow 1$	0	28	$0 \rightarrow 1$	$0 \rightarrow 1$	$1 \rightarrow 0$	48	0	0	$0 \rightarrow 1$
9	$1 \rightarrow 0$	1	0	29	1	$1 \rightarrow 0$	$1 \rightarrow 0$	49	0	0	$1 \rightarrow 0$
10	$0 \rightarrow 1$	1	0	30	$1 \rightarrow 0$	$0 \rightarrow 1$	$1 \rightarrow 0$	50	0	1	$0 \rightarrow 1$
11	$1 \rightarrow 0$	$1 \rightarrow 0$	0	31	$0 \rightarrow 1$	$1 \rightarrow 0$	$1 \rightarrow 0$	51	0	1	$1 \rightarrow 0$
12	$0 \rightarrow 1$	0	$0 \rightarrow 1$	32	1	$0 \rightarrow 1$	$1 \rightarrow 0$	52	1	1	$0 \rightarrow 1$
13	$1 \rightarrow 0$	0	$0 \rightarrow 1$	33	$1 \rightarrow 0$	1	$1 \rightarrow 0$	53	1	1	$1 \rightarrow 0$
14	0	$0 \rightarrow 1$	$0 \rightarrow 1$	34	$0 \rightarrow 1$	1	$1 \rightarrow 0$	54	1	0	$0 \rightarrow 1$
15	0	$1 \rightarrow 0$	$0 \rightarrow 1$	35	$1 \rightarrow 0$	$1 \rightarrow 0$	$1 \rightarrow 0$	55	1	0	$1 \rightarrow 0$
16	$0 \rightarrow 1$	$0 \rightarrow 1$	$0 \rightarrow 1$	36	$0 \rightarrow 1$	0	1				
17	1	$1 \rightarrow 0$	$0 \rightarrow 1$	37	$1 \rightarrow 0$	0	1				
18	$1 \rightarrow 0$	$0 \rightarrow 1$	$0 \rightarrow 1$	38	0	$0 \rightarrow 1$	1				
19	$0 \rightarrow 1$	$1 \rightarrow 0$	$0 \rightarrow 1$	39	0	$1 \rightarrow 0$	1				

Table 1. Description of each input transition for figures 2 and 3.

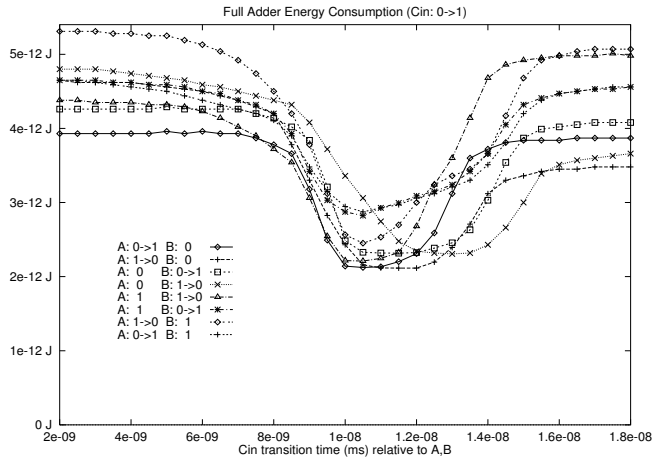

Figure 5. Power vs. Skew (C: $0 \rightarrow 1$), τ_c varied.

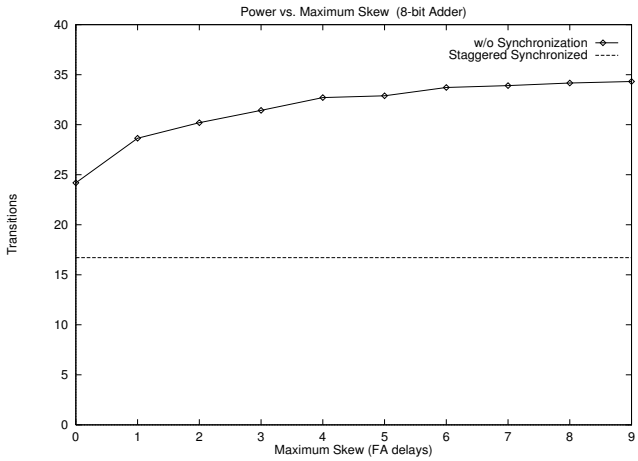

Figure 6. Effect of skew on 8-bit Carry-Ripple Adder (TSIM simulation).

sipated when the relative skew between the C_{in} input and the A or B input is somewhere between 0 and +2ns. This behavior is consistent with the timing of the cell, with the path through the circuit from C_{in} to S being shorter than the path for A or B to S.

3. Synchronization for Ripple-Carry Adder

We then measured, using TSIM [2] gate level simulation, the effects of input skew on an 8-bit ripple-carry adder. We used a simple delay model, with a unit delay for each full-adder and half-adder cell; however, we included the transitions on internal nodes of each adder cell in our transition counts. We first simulated the circuit with a randomly generated sequence of input vectors, representing the coincident arrival case. We repeated the simulation with the same input sequences, but introducing skew by delaying the arrival time of each input signal by a uniformly distributed random amount of time ($0 \leq \tau \leq \tau_{\text{max}}$). Figure 7 plots the average transitions/event vs. τ_{max}. The adder averages approximately 24 transitions per input event, or approximately 3 transitions/event/precision size. The introduction of skew increases the transition count to a limit above 34 transitions/event, or more than 4 transitions/event/precision size. This increase can be accounted for by an increase in the spurious transition activity introduced by the input arrival skew. (Note that these simulations do not take into account any spurious activity on the input signals. Such activity would further increase the power dissipation.)

We also measured the expected transition counts for the 8-bit adder using an input sequence having a "staggered

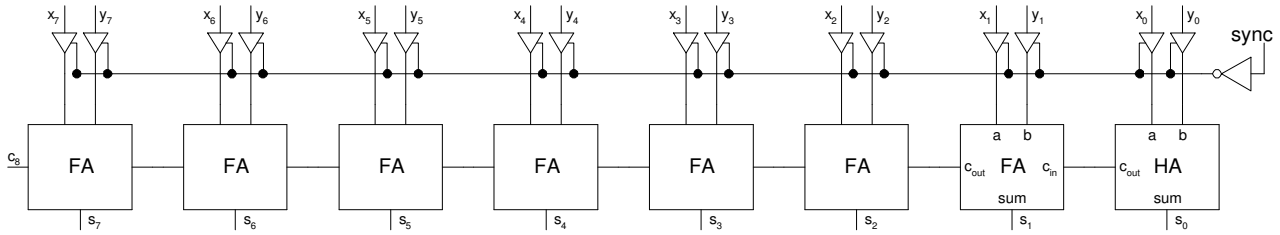

Figure 8. Ripple-Carry Adder with Input Buffers.

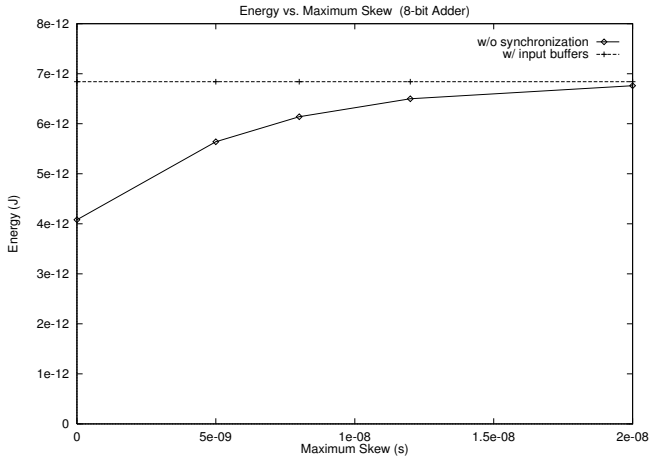

Figure 7. Effect of skew on 8-bit Carry-Ripple Adder (HSpice simulation).

skew." For each event in the sequence, the arrival time of each bit position's inputs are delayed by one time-unit from the arrival time of the inputs at the next lower bit position. Thus, the input arrival times for each bit position are synchronized to match the arrival time for the carry-in signal. The result is an average of about 16 transitions/event, or 2 transitions/event/precision size.

It is generally understood that synchronization may be used to reduce spurious transitions; however, the overhead associated with input latches may negate any benefit. Our results show that the overhead must be less than 1 transition/event/precision size in order for any power reduction to be realized.

We attempted to reduce the power consumption of the ripple-adder by adding three-state buffers to synchronize the input signals. (Figure 8.) We used HSpice to simulate this circuit and compare it with the adder without the input buffers. We repeated the simulations for different values of $\tau_{\max}$. The synchronized circuit, including overhead to generate the enable signal, dissipates about as much power as the $\tau_{\max} = 8$FA case. (1FA $\approx 5ns$).

This demonstrates the difficulty in using synchronization to reduce the transition activity. But this method is not

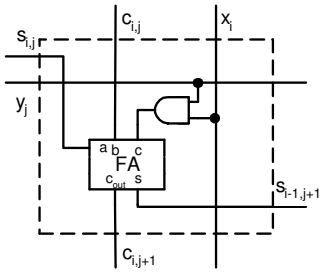

Figure 10. Partial-Product/Full-Adder (PP-FA) Cell.

without benefit:

- the input latching may be used to deactivate the circuit during periods of disuse, shielding the circuit from bus activity;
- the three-state buffers significantly reduces the input capacitance of the adder; and,
- the input latching will filter out glitches (multiple transitions) which occur on the input signals prior to their stabilization;
- less spurious activity will exist on the outputs of the full adder to be propagated to the next circuit.

4. Input Synchronization for Partial Product Reduction Arrays

We also simulated 4-bit (4x4) and 5-bit (5x5) Partial Product Reduction Arrays, again measuring the effect that random input skew has on the Power consumption. We then added three-state buffers to synchronize the latches (figure 9) and compared the results. Figure 11 plots the resulting Power vs. Skew for both arrays. With a maximum skew of $\tau_{\max} \leq$ 10ns $\approx$ 2FA, the power dissipation for the 4x4 array increases by 36% (over 40% for $\tau_{\max} \leq$ 15ns). On the other hand, the overhead for adding synchronizing three-state buffers to the circuit (including drivers for the enable signal) increases the power consumption by approximately

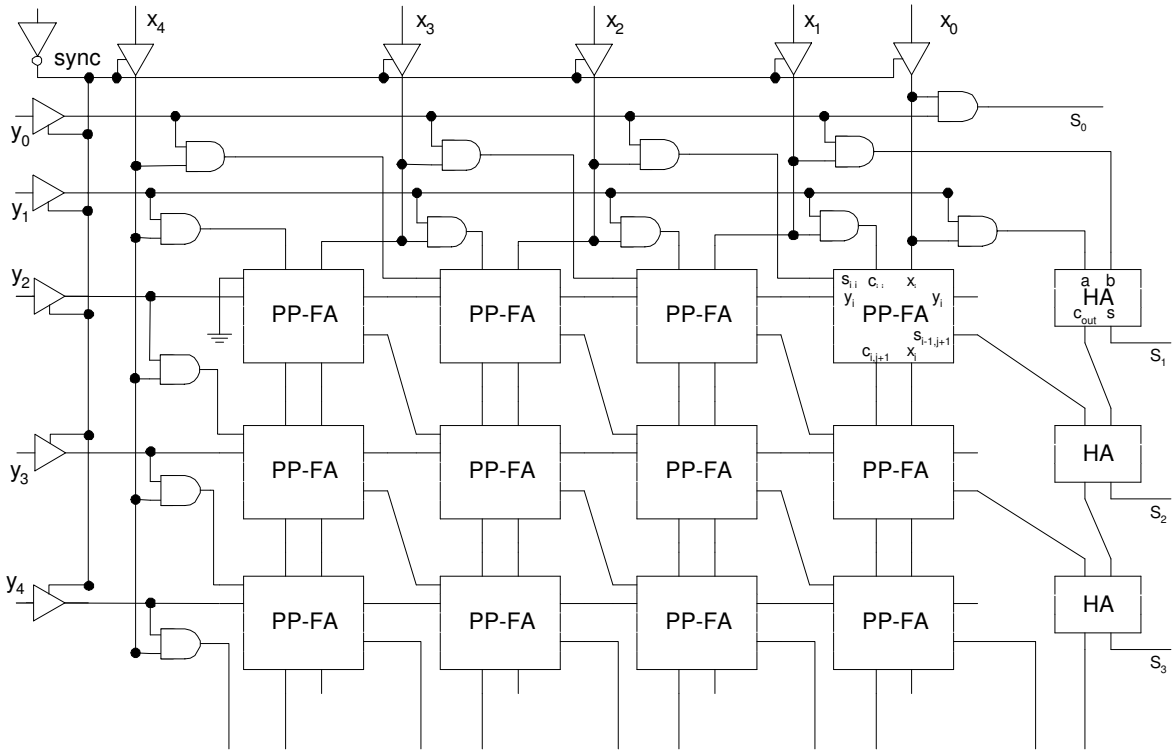

Figure 9. 5x5 Partial Product Reduction Array with Input Buffers.

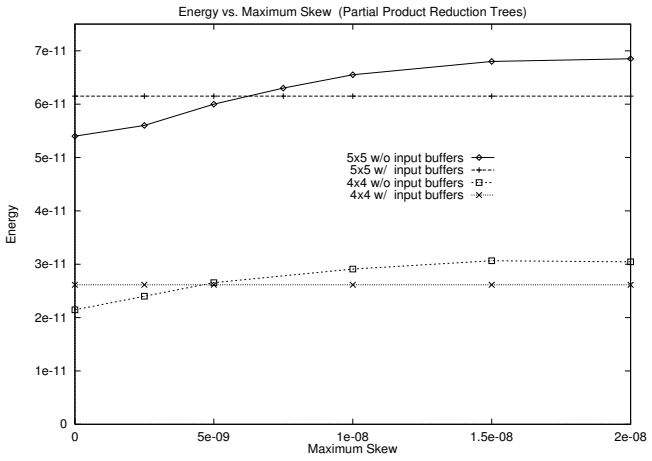

Figure 11. Effect of skew on Partial Product Reduction Arrays.

22%. Notice that for the 5x5 array, the increase in power due to skew is only 26% for the $\tau_{\max} \leq 15$ns case. Although, the overhead for adding three-state buffers is also less (14%), the break-even point for adding synchronization is at a slightly greater value of $\tau_{\max}$.

Acknowledgments. This research has been supported in part by the NSF Grant MIP-9314172 "Arithmetic Algorithms and Structures for Low-Power Systems."

References

[1] A. Chandrakasan, S. Sheng, and R. Brodersen. "Low-power CMOS digital design". *IEEE Journal of Solid-State Circuits*, 27(4):473–484, 1992.

[2] C. Fabian. TSIM manual. Technical report, UCLA Computer Science Department, Los Angeles, California, 1995.

[3] K. Keutzer and P. Vanbekbergen. "The impact of CAD on the design of low power digital circuits". In *Symposium on Low Power Electronics*, pages 42–45, San Diego, California, Oct. 1994. IEEE.

[4] U. Ko, P. T. Balsara, and W. Lee. "A self-timed method to minimize spurious transition in low power CMOS circuits". In *Symposium on Low Power Electronics*, pages 62–63, San Diego, California, Oct. 1994. IEEE.

[5] J. Leijten, J. van Meerbergen, and J. Jess. "Analysis and reduction of glitches in synchronous networks". In *Proceedings of the European Design and Test Conference*, pages 398–403, 1995.

[6] Meta-Software, Campbell, CA. *HSpice User's Manual*, 1992.

[7] S. R. Powell and P. M. Chau. "Estimating power dissipation of VLSI signal processing chips: the PFA technique". In H. S. Moscovitz, K. Yao, and R. Jain, editors, *VLSI Signal Processing, IV*, pages 250–259. IEEE Press, 1990.